8. Luana

Program Name: Luana.java **Input File: luana.dat**

Luana has just learned about a biological simulation of how cells might evolve given a certain set of rules about how they are reborn, survive, and die in several life cycle generations. In this study, a 10X10 grid is used, like the one shown below, to indicate the beginning generation. The top left cell is in position (1,1), and the one right below it is in position (2,1). Every cell has at most eight neighboring cells, one in each of the eight cardinal directions.

For this exercise, the grid shown below will be the starting generation each time a new set of evolution rules is applied. In each grid, a ***** indicates a living cell, and a **–** represents a dead cell. If the right number of living cells surrounds a dead cell, it will be reborn, or can survive to the next generation. If not enough cells surround a living cell, it will die.

Here is an example of a set of rules.

- 2 2 4: There 2 ways for a dead cell to be reborn, and that is when it is surrounded by exactly 2 or 4 living cells, otherwise it stays dead.
- 2 3 4: There are 2 ways for a living cell to survive, and that is when it has 3 or 4 living neighbors, otherwise it dies.

The research is to check the status of a particular cell, (5, 2) for example, at each point for a number of generation cycles, let's say 4.

The data set for this situation would be: 2 2 4 2 3 4 4 5 2, which shows the two rule definitions explained above, the number of generations to be evolved, and the cell to be examined.

The result will be DADAA, which means that cell (5, 2) is dead to start with, is reborn after the first generation cycle, dies after the second, is reborn after the third, and stays alive for the fourth.

Let's look at this sequence carefully, with a sample evolution of the cell in location (5,2), reborn when surrounded by 2 or 4 living cells, surviving when surrounded by 3 or 4 living cells, dying otherwise.

First generation

```
*---------
*---------
---*------
----*-----
-[-]*--*---*  Cell in (5,2) is currently dead, but is reborn in the
-*-**--*--    next generation since it is surrounded by 2 living cells
--*-----**
-*-------*
*--------*
---------*
```

Second generation

```
-*--------
-*--------
----*-----
--*--*----
-[*]-**-*-*-  Cell in (5,2) is now alive, but dies since it is only surrounded
--**-**-*-    by 2 living cells (needs 3 or 4 living neighbors to survive)
*-*-*--**-
*-*-----**
-*--------
--------*-
```

Third generation

```
*-*-------
*-*-------
-***-*----
-*-*****--
```

```
-□-**-**-*  Cell in (5,2) is dead, but is reborn in the next generation
***--**-*-   since it is surrounded by 4 living cells
--*----***
--------*-
*-*----*--
----------
```

Fourth generation

```
-*-*------
--***-----
-*-*-***--
*------*--
-*-*----*-  Cell in (5,2) is alive, and stays alive in the next generation,
-***-**--*   since it is surrounded by 3 living cells.
**-*-*-***
--**--***-
-*------*-
-*--------
```

Input: The initial 10X10 grid as described above, with a value N on the second line, followed on the next N lines with simulation data sets. Each data set consists of an integer B, followed by B integers indicating the number of neighbors required for a dead cell to be reborn, then an integer S, followed by S integers indicating the number of neighbors required for survival. After that is an integer G, which indicates how many generations to examine, and finally an ordered pair of integers R and C, indicating the cell location to study.

Output: A string of letters D or A indicating the status of the indicated cell, first showing the initial state, and the state after each generation according to the rules specified for that data set.

Sample input:

```
*---------
*---------
---*------
----*-----
--*--*---*
-*-**--*--
--*-----**
-*-------*
*--------*
---------*
2
2 2 4 2 3 4 4 5 2
3 1 2 3 3 1 2 3 2 5 10
```

Sample output:

```
DADAA
ADD
```